OPENCLASSROOMS

NOTE MÉTHODOLOGIQUE

IMPLÉMENTEZ UN MODÈLE DE SCORING

Préparer par: SIDI TEYIB BEDDY EL MOUSTAPHA

GITHUB : HTTPS://GITHUB.COM/SIDIVETE/OPENCLASSROMS_PROJET7

CONTEXTE :

Ce mémoire constitue l'un des éléments livrables du projet "Implémentez un modèle de scoring" du parcours Data Scientist d'Openclassrooms. Il expose le processus de modélisation et d'interprétabilité du modèle mis en place dans le cadre de ce projet.

L'objectif du projet est de développer un modèle de scoring de la probabilité de défaut de paiement d'un client n'ayant pas ou ayant peu d'historique de prêt, pour le compte de la société "Prêt à Dépenser", une entreprise de crédit à la consommation.

Les données utilisées pour cette tâche comprennent une base de données de 307 000 clients avec 121 caractéristiques (telles que l'âge, le sexe, l'emploi, le logement, les revenus, les informations de crédit, la notation externe, etc.).

Pour atteindre cet objectif, nous avons suivi les étapes suivantes :

MÉTHODOLOGIE D'ENTRAÎNEMENT DU MODÈLE:

L'entraînement d'un modèle constitue l'épine dorsale de l'apprentissage machine moderne, permettant aux algorithmes de devenir intelligents et performants.

Feature engineering

En raison de la dispersion et de l'incomplétude des données, cette phase implique de traiter les données de chaque tableau de manière distincte. Tout d'abord, les

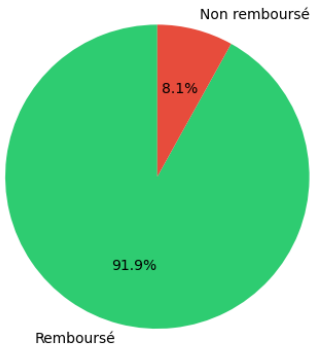
caractéristiques catégorielles sont encodées, puis des caractéristiques spécifiques au domaine sont créées à partir des données sources. Ensuite, les tableaux sont combinés pour former une seule base de données. Enfin, certaines caractéristiques sont supprimées, traiter les valeurs aberrantes et imputer les manquantes. La démarche est inspirée du kernel LightGBM with Simple Features, disponible ici : https://www.kaggle.com/code/jsaguiar/lightgbm-with-simple-features/script

Construction et entrainement des modèles

En général, un modèle de prédiction est une fonction qui prend des données en entrée et fournit en sortie une prédiction, comme dans notre cas, où il s'agit de prévoir les défauts de paiement.

Pour entraîner ces modèles de manière efficace, seule une partie des données doit être utilisée, tandis que l'autre partie est réservée pour évaluer indépendamment leurs performances. L'ensemble d'entraînement est constitué de 80% des données d'origine, tandis que l'ensemble de validation en contient 20%.

Cependant, nous faisons face à un déséquilibre dans notre échantillon, car la classe minoritaire (clients en défaut de paiement, représentés par TARGET = 1) est sous-représentée. Ce déséquilibre peut entraîner un problème de surapprentissage de la classe majoritaire (TARGET = 0).



Nous prenons donc soin de maintenir la même proportion d'échantillonnage d'origine lors de la construction des ensembles d'entraînement et de test. Cela nous permettra de traiter le déséquilibre ultérieurement en utilisant des méthodes appropriées.

Chaque modèle est soumis à un processus de validation croisée **GridSearchCV**, où chaque partie des données est utilisée alternativement comme ensemble d'entraînement et de validation. C'est à ce stade que nous appliquons le

suréchantillonnage en utilisant la **technique SMOTE** (suréchantillonnage synthétique). Cette méthode consiste à créer de nouvelles observations virtuelles pour la classe minoritaire (les individus en défaut) en se basant sur la technique des k plus proches voisins. Cela permet de donner plus d'importance aux échantillons minoritaires uniquement pendant la phase de modélisation

Nous avons testé un total de quatre modèles différents :

1. **Baseline** : Dummy Classifier
2. **Modèle classique** : Régression Logistique
3. **Forêt aléatoire** : Random Forest Classifier
4. **Boosting** : XGB Classifier

Nous avons intégré ces modèles dans mlflow tracking pour assurer un suivi et une gestion efficaces des expériences et des résultats obtenus.

LA FONCTION COÛT MÉTIER, L'ALGORITHME D'OPTIMISATION ET LA MÉTRIQUE D'ÉVALUATION

Dans le domaine professionnel, il est crucial de minimiser les coûts associés à l'identification erronée des clients susceptibles de faire défaut. Les modèles de prédiction génèrent deux types d'erreurs :

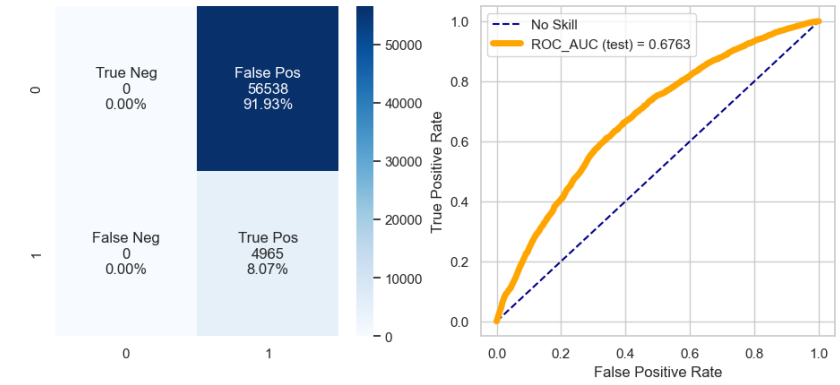
- Faux Négatif (FN) : Cela se produit lorsque le modèle identifie à tort un client comme non étant en défaut, ce qui peut entraîner l'octroi de crédits et engendrer une perte de capital.
- Faux Positif (FP) : Cela se produit lorsque le modèle identifie à tort un client comme étant en défaut, ce qui peut entraîner le refus de crédit et entraîner une perte de marge.

Il est important de noter que la perte en capital suite à un FN est environ 10 fois plus élevée que la perte de marge résultant d'un FP. Par exemple, si nous commençons avec un capital de 100 :

- FN : Perte d'environ 50% du capital prêté en moyenne, soit une perte de 50.
- FP : Manque à gagner d'environ 1% par an sur 10 ans en moyenne, ce qui représente une perte de 5 (10% de 50).

L'objectif est de minimiser à la fois les FN et les FP pour éviter ces fausses prédictions. Pour cela, nous souhaitons réduire le score de la fonction (10 * FN + FP) / taille de l'échantillon (un ratio entre 0 et 1) issu de la matrice de confusion, qui compare les prévisions du modèle aux valeurs réelles pour chaque modèle testé. Ce score est utilisé pour évaluer les modèles dans l'étape de GridSearchCV et, finalement, pour les sélectionner. D'autres scores tels que le temps, le ROC_AUC, etc., sont également calculés pour évaluer les performances des modèles.

Pour chaque modèle sélectionné par le GridSearchCV, nous avons réalisé une optimisation du seuil pour déterminer la probabilité minimale de défaut à partir de laquelle une observation doit être classée en tant que défaut. Avant cette optimisation, le seuil par défaut est fixé à 0,5. Cette opération est réalisée à l'aide d'une boucle qui parcourt 100 valeurs comprises entre 0 et 1. L'objectif est de trouver la valeur du seuil pour laquelle la fonction de coût métier (10 FN + FP) / taille de l'échantillon est minimale. Cette approche nous permet d'ajuster le seuil de classification de manière à optimiser les performances du modèle en termes de coûts de fausses prédictions.



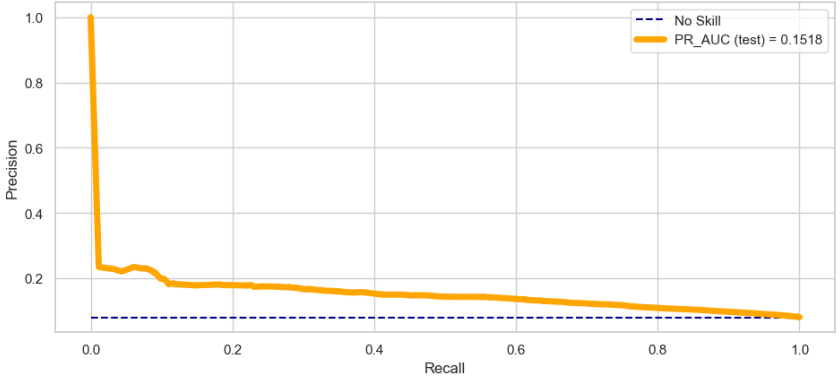
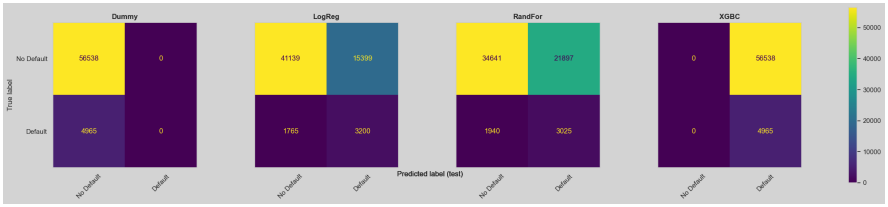
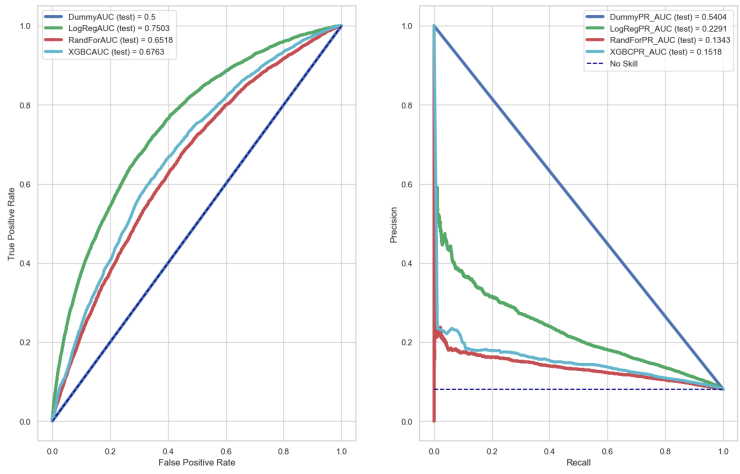


TABLEAU DE SYNTHÈSE DES RÉSULTATS

	Data	Model	Custom_score	ROC_AUC	PR_AUC
0	Train	Dummy	0.8073	0.5	0.540365
1	Test	Dummy	0.8073	0.5	0.540364
2	Train	LogReg	0.5384	0.7508	0.223985
3	Test	LogReg	0.5374	0.7503	0.229102
4	Train	RandFor	0.561	0.7397	0.184874
5	Test	RandFor	0.6715	0.6518	0.134322
6	Train	XGBC	0.561	0.7397	0.184874
7	Test	XGBC	0.9193	0.6763	0.151798

D'après les résultats obtenus sur les tables, le modèle XGB (XGBoost) obtient le meilleur custom score, c'est-à-dire qu'il minimise le coût métier. En outre, il présente les plus grandes aires sous la courbe ROC_AUC (Receiver Operating Characteristic Area Under Curve) et PR_AUC (Precision-Recall Area Under Curve). Cela indique que le modèle XGBoost est non seulement performant en termes de coût métier, mais également en termes de capacité à discriminer les classes et à maintenir un bon équilibre entre la précision et le rappel dans ses prédictions.





L'INTERPRÉTABILITÉ GLOBALE ET LOCALE DU MODÈLE

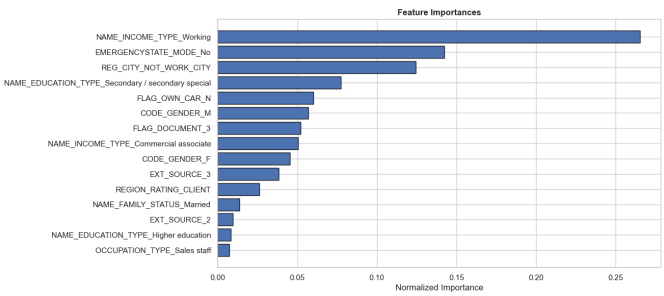
Le modèle que nous développons est destiné à être utilisé par des professionnels de la banque lorsqu'ils traitent avec leurs clients. Il est essentiel que les décisions d'octroi de crédit soient expliquées de manière claire et transparente.

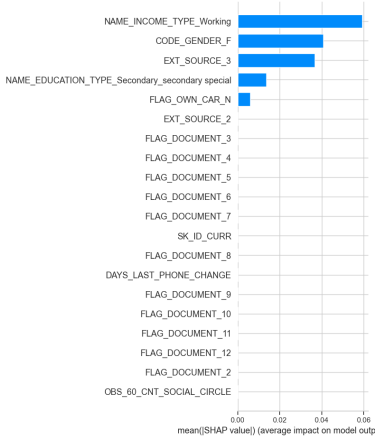
Cependant, les modèles de prédiction ont tendance à agir comme des "boîtes noires", rendant difficile la compréhension des relations de cause à effet qui sous-tendent leurs prédictions.

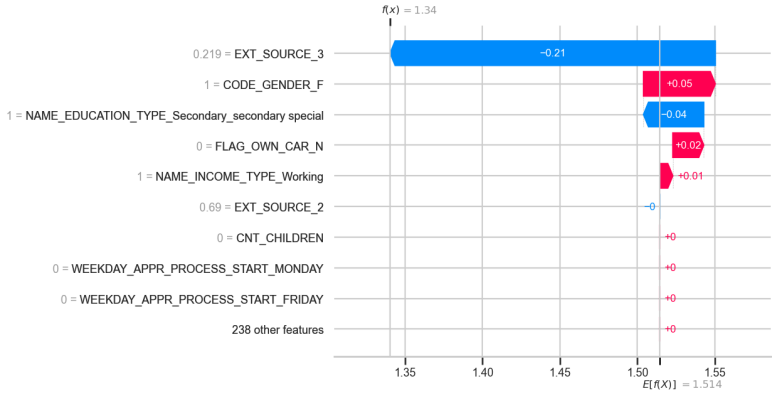
Pour surmonter cette limitation, nous calculons les "Shap values" (SHapley Additive exPlanations), qui représentent l'impact supplémentaire de chaque caractéristique dans la prédiction de chaque probabilité.

Grâce à cette approche, il devient possible d'expliquer non seulement globalement le poids de chaque caractéristique dans les prédictions du modèle, mais aussi de comprendre localement le poids de chaque caractéristique dans la prédiction associée à une observation spécifique. Cela se fait en partant de la valeur de base attendue et en sommant les "SHAP values" jusqu'à obtenir la probabilité prédite par le modèle.

En utilisant les "Shap values", nous sommes en mesure de fournir des explications claires et compréhensibles pour chaque décision d'octroi de crédit, permettant ainsi une meilleure transparence et une justification des choix effectués par le modèle







LES LIMITES ET LES AMÉLIORATIONS POSSIBLES

- 1- Dans le contexte financier, il est préférable d'explorer d'autres modèles plus rapides et compatibles avec le domaine, tels que LightGBM, en vue d'améliorer l'efficacité et les performances du système.

- 2- Le choix du meilleur modèle repose entièrement sur la fonction coût métier. Cependant, il est important de valider et d'éventuellement améliorer les hypothèses sous-jacentes à la construction de cette fonction en consultant les professionnels du domaine financier.

- 3- Étant donné que le pré-traitement des données n'a pas subi de modifications significatives et n'a pas été remis en question dans le cadre de ce projet, il est probable qu'un travail plus approfondi de nettoyage et d'ingénierie des caractéristiques pourrait conduire à de meilleurs résultats en termes de qualité des prédictions. Une analyse minutieuse et des ajustements adéquats des données pourraient ainsi améliorer la performance globale du modèle.

L'ANALYSE DU DATA DRIFT

Pour évaluer la stabilité et la cohérence des données, nous avons réalisé une analyse de datadrift. En examinant les résultats, nous avons observé les constats suivants.**Entre X_train, Y_train et X_test, Y_predict** : Nous avons constaté des drifts dans 9 colonnes.Ces différences entre les ensembles de données peuvent indiquer des variations dans les distributions des caractéristiques et des étiquettes entre les différents jeux de données. L'analyse du datadrift nous aide à repérer ces déviations potentielles, ce qui est important pour assurer la fiabilité et la performance des modèles de prédiction lorsqu'ils sont déployés dans des environnements réels. Il convient d'examiner attentivement ces drifts afin de déterminer s'ils peuvent être attribués à des changements naturels dans les données ou s'ils nécessitent des ajustements ou des corrections dans les processus de collecte et de traitement des données.